REMARKS

Amendments to claims 1, 11, and 14 have been requested in order to address minor 35 U.S.C. §112 issues and do not raise any new issues. It is requested that it be entered if the case is not allowed for purposes of appeal under 37 CFR §1.116.

Plug and play devices that facilitate a seamless connect and disconnect capability with a controlling device are becoming ubiquitous. One common problem associated with the use of these plug and play devices is that frequently the master device must load controlling software when a peripheral device is plugged in. This takes time, producing an inherent latency when a plug and play device is introduced and the controlling software is loaded. Innovations that eliminate or reduce this latency can improve the usability of many plug and play systems.

The present invention mitigates the effect of this latency problem by eliminating the need to reload controlling software under certain predetermined conditions and reducing the reload time under other conditions.

One of the times this latency manifests itself is when a device is temporarily unplugged or becomes unavailable. This can occur when a user inadvertently disconnects a device, reconfigures an existing device or a power transient occurs. When a peripheral device temporarily becomes unavailable many master devices will uninstall some or all of the requisite control software to clear memory for other functions. When the user reconnects the peripheral device the master device must then reload the necessary control software.

In many system architectures, reloading control software may place a substantive burden on the system and the user. In some systems, the device will reload the software

via the Internet or an external media, often taking considerable time and system resources.

The applicant's invention solves this problem in part by the introduction of a timing control unit that controls and develops an uninstall function of peripheral device software. The timing control unit, upon detection that a device has been unplugged or is unavailable, will start a timer to provide a predetermined time period of a sufficient duration to enable a user or a bus reset to cause the system to re-connect the peripheral device. The timer will delay the uninstall process for this predetermined time period thereby allowing a grace period to enable a user to reconnect a device, reconfigure a device, or allow the system to recover to a quiescent state after a power transient as set forth on Page 5, Lines 8-15 of our specification:

> With this construction, the uninstall process is started after the predetermined time has passed since the detection unit detected the disconnection of the slave device. Therefore, even in the case where a user accidentally disconnects the slave device, the user can use the slave device again without delay simply by reconnecting the slave device to the system before the predetermined time has passed.

Alternatively, our invention can also address situations where the master device cannot detect slave devices in response to generation of a bus reset signal.

In addition, the timing control unit may also feature intelligent de-install control software to decrease the reinstall burden if the same peripheral device is reintroduced after the de-installation process has begun.

The *Maeda et al* storage unit teaches an apparatus for managing drivers for peripheral devices. The *Maeda et al* disclosure describes how to detect the connection of

a peripheral device [Paragraph 44] and disconnection of a peripheral device [Paragraph 47]. It describes an apparatus that implements the plug and play functionality.

The *Maeda et al* disclosure is very detailed in its disclosure about the detection of the connection and disconnection of peripheral devices. Figure 5 shows a delay from the time a plug and play device is connected or disconnected and the time the controller detects the event. This time delay is not fixed but is a function of the resistor R1 and wire capacitance of the data cable [Paragraph 48]. What *Maeda et al* is describing in this figure is a hardware delay in the detection of the connection or removal of a peripheral device.

Although the *Maeda et al* disclosure uses timing diagrams to describe the operation of its system it does not teach the use of a timing means to delay uninstalling control software to prevent the need to re-install software. Thus the *Maeda et al* disclosure does not disclose an apparatus capable of mitigating plug and play latency problems by delaying the de-installation of control software.

The Office Action cited a timing control means for uninstalling software in Column 8, Lines 20-37 of *Maeda et al*. However, the cited passage actually teaches away from applicant's timing control means which ***delays*** the ***de-installation of control software*** to allow the user the opportunity to reconnect a disconnected device without the need to reinstall the control software.

The cited passage describes a pseudo operation of the controller. In this mode the controller regulates power supply to emulate disconnect and reconnect of a peripheral device in order ***to trigger install*** of new device drivers. In other words, the controller cycles power to a peripheral device to initiate a control software load.

> [I]t is generally settled that the change in prior art device which makes the device inoperable for its intended purpose cannot be considered to be an obvious change.
>
> *Hughes Aircraft Co. v. United States*, 215 U.S.P.Q. 787, 804 (Ct. Cl. Trial Div. 1982)

Furthermore, neither the passage or the timing diagrams show any delay between the PC detection of the device disconnect and the beginning of a driver de-installation process. The elaborate timing diagrams showing RC voltage decay and ramp up curves are not design features but a truism of a DC circuit.

The Office Action rejected Claims 1-3, 9-12 and 14-15 over *Maeda et al* (EPO 905608) under 35 U.S.C. §102.

The Office Action rejected Claim 1, specifically pointing out that the *Maeda et al* reference teaches the use of elapsed time intervals. The Office Action cites a passage [Column 8, 20-37] that describes the timing diagram of the controller in pseudo-operation [Column 8, 16-20]. The time periods, T2 and T3 cited in the Office Action passage described show the hardware detection delay and the real time sequential operations of the *Maeda et al* controller, respectively.

For example, looking only at the timing diagram and the cited passage, a reader might infer the controller 205 also measures the elapsed time (T3) from the detection of disconnection. However, T3 is the "time in which PC 102 can delete the driver for the device and rearrange the inside of the system." The time T3 is not a predetermined value used by the controller, rather T3 is the amount of time it happens to take the PC to complete the driver uninstall the driver.

Further highlighting the lack of a controller induced delay or the use of an elapsed time interval in the *Maeda et al* disclosure is the flow diagram in Figure 9. In the flow diagram upon detection of a device disconnection the PC begins the uninstall process (S901-S902). This immediate flow from the detection of disconnect to the clearing the driver from memory is emphasized in the disclosure (Column 11, Lines 11-16). The controller does not delay the process of de-installation. The PC detects the disconnection device by a drop in voltage and then begins the de-installation process.

It also should be noted that T2 waveform is also not a predetermined time delay as claimed. T2 shown in Figure 5 is the time delay from actual disconnect to the time the voltage drop is large enough for the PC to detect disconnect. This time is highly dependent on the operating environment. T2 will depend among other things on the threshold voltage of the PC, the input impedance of the PC, the cable capacitance, and individual resistor values. In addition, T2 is likely to be very short in duration and far too short to allow a user to reconnect a device.

Claims 2-3, 9-12 and 14-15 are all dependent on Claim 1 and are novel based on the reasoning for Claim 1.

It should also be noted that the Claim 3 also differs from the *Maeda et al* disclosure in that it controls the de-installation process by breaking it into multiple stages. This intelligent de-installation reduces the reinstallation time if a plug and play device is reintroduced during the process.

In contrast the *Maeda et al* disclosure describes a time T3 in which the PC can delete the driver for the device [Column 9, line 35]. The Office Action points out that this process may have been initiated at an earlier time. The time the Office Action is

referring is presumably T2 the time from when the controller initiates a power interruption until the time the PC detects it. This RC time constant decay is not a multi-stage de-installation as claimed in claim 3.

Claims 11-13 are method claims for the apparatus described in Claims 1-4 respectively and are novel based on the same reasoning.

The computer readable recording medium Claims 14-16 correspond to the apparatus described in claims 1-4 and are novel based on the same reasoning.

Claims 4, 6-8, 13, 16 and 18 were rejected over the *Maeda et al* reference in view of the *Davis* (U.S. Patent No. 5,862,393) under 35 U.S.C. §103.

However, the *Davis* reference teaches a restoration means for restoring data modified during an uninstall process. The *Davis* disclosure teaches a power management system that will advise a devise driver that a devise power down event will occur so that the devise driver can store configuration data before power down. This driver will then have the configuration of the device when the power management system brings the device back on-line.

The *Davis* disclosure does not, however, teach waiting a predetermined time before uninstalling the device to allow reinsertion of a peripheral device. To the contrary, the *Davis* disclosure teaches a power management system that simulates the removal of a peripheral device to save power. Neither the *Davis* disclosure nor the *Maeda et al* reference teaches the use of a predetermined time interval to delay the de-installation of control software.

Claims 5 and 17 were rejected over a combination of *Maeda et al*, *Davis* and the *Danforth* (U.S. Patent No. 5,493,680.

REMARKS

Amendments to claims 1, 11, and 14 have been requested in order to address minor 35 U.S.C. §112 issues and do not raise any new issues. It is requested that they be entered if the case is not allowed for purposes of appeal under 37 CFR §1.116.

Plug and play devices that facilitate a seamless connect and disconnect capability with a controlling device are becoming ubiquitous. One common problem associated with the use of these plug and play devices is that frequently the master device must load controlling software when a peripheral device is plugged in. This takes time, producing an inherent latency when a plug and play device is introduced and the controlling software is loaded. Innovations that eliminate or reduce this latency can improve the usability of many plug and play systems.

The present invention mitigates the effect of this latency problem by eliminating the need to reload controlling software under certain predetermined conditions and reducing the reload time under other conditions.

One of the times this latency manifests itself is when a device is temporarily unplugged or becomes unavailable. This can occur when a user inadvertently disconnects a device, reconfigures an existing device or a power transient occurs. When a peripheral device temporarily becomes unavailable many master devices will uninstall some or all of the requisite control software to clear memory for other functions. When the user reconnects the peripheral device the master device must then reload the necessary control software.

In many system architectures, reloading control software may place a substantive burden on the system and the user. In some systems, the device will reload the software

via the Internet or an external media, often taking considerable time and system resources.

The applicant's invention solves this problem in part by the introduction of a timing control unit that controls and delays an uninstall function of peripheral device software. The timing control unit, upon detection that a device has been unplugged or is unavailable, will start a timer to provide a predetermined time period of a sufficient duration to enable a user or a bus reset to cause the system to re-connect the peripheral device. The timer will delay the uninstall process for this predetermined time period thereby allowing a grace period to enable a user to reconnect a device, reconfigure a device, or allow the system to recover to a quiescent state after a power transient as set forth on Page 5, Lines 8-15 of our specification:

> With this construction, the uninstall process is started after the predetermined time has passed since the detection unit detected the disconnection of the slave device. Therefore, even in the case where a user accidentally disconnects the slave device, the user can use the slave device again without delay simply by reconnecting the slave device to the system before the predetermined time has passed.

Alternatively, our invention can also address situations where the master device cannot detect slave devices in response to generation of a bus reset signal.

In addition, the timing control unit may also feature intelligent de-install control software to decrease the reinstall burden if the same peripheral device is reintroduced after the de-installation process has begun.

The *Maeda et al* storage unit teaches an apparatus for managing drivers for peripheral devices. The *Maeda et al* disclosure describes how to detect the connection of

a peripheral device [Paragraph 44] and disconnection of a peripheral device [Paragraph 47]. It describes an apparatus that implements the plug and play functionality.

The *Maeda et al* disclosure is very detailed in its disclosure about the detection of the connection and disconnection of peripheral devices. Figure 5 shows a delay from the time a plug and play device is connected or disconnected and the time the controller detects the event. This time delay is not fixed but is a function of the resistor R1 and wire capacitance of the data cable [Paragraph 48]. What *Maeda et al* is describing in this figure is a hardware delay in the detection of the connection or removal of a peripheral device.

Although the *Maeda et al* disclosure uses timing diagrams to describe the operation of its system it does not teach the use of a timing means to delay uninstalling control software to prevent the need to re-install software. Thus the *Maeda et al* disclosure does not disclose an apparatus capable of mitigating plug and play latency problems by delaying the de-installation of control software.

The Office Action cited a timing control means for uninstalling software in Column 8, Lines 20-37 of *Maeda et al*. However, the cited passage actually teaches away from applicant's timing control means which ***delays*** the ***de-installation of control software*** to allow the user the opportunity to reconnect a disconnected device without the need to reinstall the control software.

The cited passage describes a pseudo operation of the controller. In this mode the controller regulates power supply to emulate disconnect and reconnect of a peripheral device in order ***to trigger install*** of new device drivers. In other words, the controller cycles power to a peripheral device to initiate a control software load.

> [I]t is generally settled that the change in prior art device which makes the device inoperable for its intended purpose cannot be considered to be an obvious change.
>
> *Hughes Aircraft Co. v. United States*, 215 U.S.P.Q. 787, 804 (Ct. Cl. Trial Div. 1982)

Furthermore, neither the passage or the timing diagrams show any delay between the PC detection of the device disconnect and the beginning of a driver de-installation process. The elaborate timing diagrams showing RC voltage decay and ramp up curves are not design features but a truism of a DC circuit.

The Office Action rejected Claims 1-3, 9-12 and 14-15 over *Maeda et al* (EPO 905608) under 35 U.S.C. §102.

The Office Action rejected Claim 1, specifically pointing out that the *Maeda et al* reference teaches the use of elapsed time intervals. The Office Action cites a passage [Column 8, 20-37] that describes the timing diagram of the controller in pseudo-operation [Column 8, 16-20]. The time periods, T2 and T3 cited in the Office Action passage described show the hardware detection delay and the real time sequential operations of the *Maeda et al* controller, respectively.

For example, looking only at the timing diagram and the cited passage, a reader might infer the controller 205 also measures the elapsed time (T3) from the detection of disconnection. However, T3 is the "time in which PC 102 can delete the driver for the device and rearrange the inside of the system." The time T3 is not a predetermined value used by the controller, rather T3 is the amount of time it happens to take the PC to complete the driver uninstall the driver.

Further highlighting the lack of a controller induced delay or the use of an elapsed time interval in the *Maeda et al* disclosure is the flow diagram in Figure 9. In the flow diagram upon detection of a device disconnection the PC begins the uninstall process (S901-S902). This immediate flow from the detection of disconnect to the clearing the driver from memory is emphasized in the disclosure (Column 11, Lines 11-16). The controller does not delay the process of de-installation. The PC detects the disconnection device by a drop in voltage and then begins the de-installation process.

It also should be noted that T2 waveform is also not a predetermined time delay as claimed. T2 shown in Figure 5 is the time delay from actual disconnect to the time the voltage drop is large enough for the PC to detect disconnect. This time is highly dependent on the operating environment. T2 will depend among other things on the threshold voltage of the PC, the input impedance of the PC, the cable capacitance, and individual resistor values. In addition, T2 is likely to be very short in duration and far too short to allow a user to reconnect a device.

Claims 2-3, 9-12 and 14-15 are all dependent on Claim 1 and are novel based on the reasoning for Claim 1.

It should also be noted that the Claim 3 also differs from the *Maeda et al* disclosure in that it controls the de-installation process by breaking it into multiple stages. This intelligent de-installation reduces the reinstallation time if a plug and play device is reintroduced during the process.

In contrast the *Maeda et al* disclosure describes a time T3 in which the PC can delete the driver for the device [Column 9, line 35]. The Office Action points out that this process may have been initiated at an earlier time. The time the Office Action is

referring is presumably T2 the time from when the controller initiates a power interruption until the time the PC detects it. This RC time constant decay is not a multi-stage de-installation as claimed in claim 3.

Claims 11-13 are method claims for the apparatus described in Claims 1-4 respectively and are novel based on the same reasoning.

The computer readable recording medium Claims 14-16 correspond to the apparatus described in claims 1-4 and are novel based on the same reasoning.

Claims 4, 6-8, 13, 16 and 18 were rejected over the *Maeda et al* reference in view of the *Davis* (U.S. Patent No. 5,862,393) under 35 U.S.C. §103.

However, the *Davis* reference teaches a restoration means for restoring data modified during an uninstall process. The *Davis* disclosure teaches a power management system that will advise a devise driver that a devise power down event will occur so that the devise driver can store configuration data before power down. This driver will then have the configuration of the device when the power management system brings the device back on-line.

The *Davis* disclosure does not, however, teach waiting a predetermined time before uninstalling the device to allow reinsertion of a peripheral device. To the contrary, the *Davis* disclosure teaches a power management system that simulates the removal of a peripheral device to save power. Neither the *Davis* disclosure nor the *Maeda et al* reference teaches the use of a predetermined time interval to delay the de-installation of control software.

Claims 5 and 17 were rejected over a combination of *Maeda et al*, *Davis* and the *Danforth* (U.S. Patent No. 5,493,680.

The *Danforth* reference was cited for its teaching of object-oriented languages and more specifically, a statement that unloads a class file and removes the class file from a System Object Model. The *Danforth* reference does not provide the determination of a predetermined time from a disconnect nor provide a comparison with that time that prohibits a de-installation to permit a possibility of a re-connect.

It is believed that the present application is in condition for allowance and an early notification is requested.

If there are any concerns with regard to this prosecution, or if the Examiner believes that a telephone interview will help further prosecution of the case, he is respectfully requested to contact the undersigned attorney at the listed telephone number.

I hereby certify that this correspondence is being transmitted via facsimile to the USPTO at **571-273-8300** on September 16, 2005.

By: Sharon Farnus
Sharon Farnus
Signature

Dated: September 16, 2005

Very truly yours,

SNELL & WILMER L.L.P.

Joseph W. Price
Registration No. 25,124
600 Anton Boulevard, Suite 1400
Costa Mesa, California 92626-7689
Telephone: (714) 427-7420